AFRL-AFOSR-JP-TR-2016-0030

Framework for Processing Videos in the Presence of Spatially Varying Motion Blur

Ambasamudram Rajagopalan
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

02/10/2016
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ IOA
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved

OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
23-06-2016	Final	30 Sep 2013 - 29 Sep 2015

4. TITLE AND SUBTITLE

Framework for Processing Videos in the Presence of Spatially Varying Motion Blur

5a. CONTRACT NUMBER

FA23861314138

5b. GRANT NUMBER

13RSZ116_134138

5c. PROGRAM ELEMENT NUMBER

61102F

6. AUTHOR(S)

Prof Ambasamudram Narayanan Rajagopalan

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Indian Institute of Technology Madras
IIT Madras
Chennai 600036
India

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AOARD
UNIT 45002
APO AP 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/AFOSR/IOA(AOARD)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AOARD-134138

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution A: Approved for public release. Distribution is unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Motion blurring is both a bane and a boon. Most works treat motion blur as nuisance and seek ways and means to mitigate its effects so as to restore the original image. Unlike the optical blur, motion-blur can be space-varying even when the scene is planar; an example case is that of a rotating camera imaging a distant scene. However, it must be emphasized that motion blur can also serve as a vital cue for camera motion estimation, depth recovery, super-resolution, and image forensics.

15. SUBJECT TERMS

full motion video analysis, Image Processing, Video analysis, Information Technology

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Seng Hong, Ph.D.
U	U	U	SAR	41	19b. TELEPHONE NUMBER *(Include area code)* +81-3-5410-4409

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

FINAL REPORT FOR AOARD-AFRL

(FA2386-13-1-4138, Year 2013-2014)

TITLE	**Framework for Processing Videos in the Presence of Spatially Varying Motion Blur**
PI	Prof. A.N. Rajagopalan, Indian Institute of Technology Madras
AFRL POC	Dr. Guna Seetharaman, Civ DR-IV AFRL/RIEA
AOARD PM	Dr. Seng Hong
AFOSR PM	Dr. Tristan Nguyen, AFOSR/RSL
Duration	Oct. 1. 2013 - September 30, 2014
Cost	50K (FY14)

1 Introduction

The current proposal is focused on a basic (6.1) level research on full motion video analysis - a topic of importance to the U.S. Air Force - with a potential impact on image analysis, characterization and exploitation. The amount of full motion video clips that we process has grown exponentially. These images are typically acquired for surveillance purpose, collected persistently over a fixed field of view, albeit with varying degrees of relative motion between the camera and objects within the scene. The key challenge is to handle the complexities (including loss of resolution) that arise from space-varying local blurring due to camera motion.

Recent times have seen the resurgence of motion blur as an area of great interest to computer vision and image processing researchers. Motion blur results when there is relative motion between the camera and the scene. For planar scenes, the shape of the blur kernel is a function of camera motion while the weights of the kernel can be related to the exposure time corresponding to the set of geometric transformations that the camera traversed along its motion trajectory. Motion blur has acquired special significance with hand-held imaging, aerial imaging, and imaging 'on the move' shooting into prominence. It is also relevant to situations where the camera is still but the scene comprises of several moving objects.

Motion blurring is both a bane and a boon. Most works treat motion blur as nuisance and seek ways and means

to mitigate its effects so as to restore the original image. Unlike the optical blur, motion-blur can be space-varying even when the scene is planar; an example case is that of a rotating camera imaging a distant scene. However, it must be emphasized that motion blur can also serve as a vital cue for camera motion estimation, depth recovery, super-resolution, image forensics, etc.

In this report, we discuss the efforts carried out jointly with Dr. Guna Seetharaman during the period Oct. 1, 2013 to Sept. 30, 2014. There was regular exchange of information between the PI and the AFRL collaborator including physical meetings along the sidelines of conferences. We first invesgated the recovery of normal of a planar sccene imaged by a moving camera. Here, the motion blur is harnessed as a cue for estimation of normal since the extent of motion blur at an image point is dictated both by scene structure and camera motion. We have developed a scheme for recovering the orientation of a planar scene from a single translationally-motion blurred image. By leveraging the homography relationship among image coordinates of 3D points lying on a plane, and by exploiting natural correspondences among the extremities of the blur kernels derived from the motion blurred observation, the proposed method can accurately infer the normal of the planar surface. We validate our approach on synthetic as well as real planar scenes.

Next, we addressed the problem of image registration using low-rank, sparse error matrix decomposition when there are geometric as well as photometric differences in the given image pair. The additional challenge is to perform registration and change detection for large motion blurred images. The unreasonable demand that this task puts on computational and memory resources precludes the possibility of any direct attempt at solving this problem. We handle this issue by observing the fact that the camera motion experienced by a sufficiently large sub-image is approximately the same as that of the entire image itself. We devise an algorithm for judicious sub-image selection so that the camera motion can be deciphered correctly, irrespective of the presence or absence of occluder. We adopt a reblur-difference framework to detect changes as this is an artifact-free pipeline unlike the traditional deblur-difference approach. We demonstrate the results of our algorithm on both synthetic and real data.

Following this, we attempt to solve the problem of motion deblurring which has significant ramifications in aerial imaging. Our work deals with deblurring of aerial imagery and we develop a methodology for blind restoration of spatially varying blur induced by camera motion caused by instabilities of the moving platform. A sharp image is beneficial not only from the perspective of visual appeal but also because it forms the basis for applications such as moving object tracking, change detection, and robust feature extraction. In the presence of general camera motion, the apparent motion of scene points in the image will vary at different locations resulting in space-variant blurring.

However, due to the large distances involved in aerial imaging, we show that the blurred image of the ground plane can be expressed as a weighted average of geometrically warped instances of the original focused but unknown image. The weight corresponding to each warp denotes the fraction of the total exposure duration the camera spent in that pose. Given a single motion blurred aerial observation, we propose a scheme to estimate the original focused image affected by arbitrarily-shaped blur kernels. The latent image and its associated warps are estimated by optimizing suitably derived cost functions with judiciously chosen priors within an alternating minimization framework. Several results are given on the challenging VIRAT aerial dataset for validation.

In the following sections, we discuss each of the above problems in more detail. **While some of the results of these efforts have already been published in IEEE conferences and journals, others are under review in prestigious venues.**

2 Normal Inference from a Single Motion Blurred Image

An extensively researched area in computer vision is the recovery of 3D structure from image intensities [1]. Well-known cues for depth recovery include disparity, optical flow, texture, shading, defocus blur and motion blur, to name a few. While estimation of 3D depth/shape has been of general interest, there have also been works targeting the special case of inferring planar 3D geometry (such as the Manhattan model). This is due to the fact that the world around us can, in many cases, be modeled as being piecewise planar. Approximating a 3D scene with planes (where possible) has tremendous advantage in terms of reducing the computational complexity. Estimation of surface normals of a scene/object plays a crucial role in identifying the 3D geometry/shape of that scene/object. The elegant homography relationship between two images (original and transformed due to relative motion between camera and scene) holds for scene points lying on a plane in the 3D world.

Estimating a plane involves finding its surface normal and the perpendicular distance from the center of the camera to the plane. The relevance of this problem is evident from the many works that exist in the literature. Clark et al. [2] implemented a technique to recover the orientation of text planes using perspective geometry. In [3], Farid et al. reveal the fact that the projection of a planar texture having random phase leads to higher-order correlations in the frequency domain, and these correlations are proportional to the orientaion of the plane. Greinera et al. [4] have proposed a method to determine the surface normal using projective geometry and spectral analysis. Haines et al. [5] describe a technique that makes use of prior training data gathered in an urban environment to classify planar/non-planar surfaces

and to compute the orientaion of the planes.

We propose to use motion blur as a *cue* to estimate the *orientation* of a planar scene given a *single* motion blurred image of the plane. Usually, blurring is considered as a nuisance whose effect needs to be removed. However, works do exist that, in fact, use blur (optical/motion) as a cue to infer valuable information such as the depth of the scene and relative motion of the camera with respect to the scene.

To the best of our knowledge, the only method to estimate plane orientation using blur as a cue is the recent work by McCloskey et al. [6] who have proposed a method based on blur gradients to evaluate the planar orientaion (slant and tilt angles) from a single image using *optical blur* as a cue. They exploit the relationship between blur variations for the equifocal (fronto-parallel scene) plane and a plane's tilt and slant angles. For a fronto-parallel scene, all the pixels in the image have the same amount of blur. In the case of an inclined plane, the amount of blur varies inversely with depth. The user has to manually mark a patch of interest for which slant and tilt angles are estimated. Their work assumes a homogeneously textured observation.

We propose an interesting approach (a first of its kind) to determine the surface normal of a plane from a *single motion-blurred* image. We exploit the homography relation that exists in the image domain under camera motion to determine the surface normal. For a planar scene, the blurred image can be represented as a weighted average of warped versions of the unblurred image. This representation helps in characterizing the space-variant blur by a set of global homographies. We extract patches from the image and estimate blur kernels at these patches. Using the correspondences among the extremities of blur kernels at different locations, we set up a system of linear equations that are solved to yield the surface normal.

2.1 Planar motion blur

Motion blur in an image is due to relative motion between camera and scene during exposure time. Since the camera sensor sees different scene points at different instants of time within the exposure window, these intensities get averaged resulting in a blurred image. Let g be the blurred image captured by a camera with exposure time E_t, and let f be the original image (without camera shake). During the exposure time, f may have undergone a set of transformations due to relative motion between the camera and the scene. The transformed image at time instant τ can be explained using homography H_τ as $g_\tau(H_\tau(\mathbf{x})) = f(\mathbf{x})$ where $\mathbf{x}$ represents pixel coordinates. Therefore, the blurred image can be modeled as the average of transformed versions of f during the exposure time E_t. The blurred image intensity at

a location $\mathbf{x}$ can then be expressed as

$$g(\mathbf{x}) = \frac{1}{E_t} \int_0^{E_t} f(H_\tau^{-1}(\mathbf{x}))\, d\tau$$

The homography relation in the image domain holds only for the set of scene points lying on a plane. The homography at time instant τ is given by $H_\tau = K\left(R_\tau + \mathbf{t}_\tau \frac{\mathbf{n}^T}{d}\right) K^{-1}$ where $K = \begin{bmatrix} q & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & 1 \end{bmatrix}$, with q being the focal length of the camera. Here R_τ denotes the rotation matrix at time instant τ and is a combination of the rotational matrices about the X, Y and Z axes, and d is the perpendicular distance from the center of the camera to the plane and is a constant for the entire plane. Here, $\mathbf{t}_\tau = [T_{X_\tau} T_{Y_\tau} T_{Z_\tau}]^T$ represents the 3D translation vector at time τ and $\mathbf{n} = [N_X\ N_Y\ N_Z]^T$ denotes the surface normal of the planar scene. Following recent works, we assume that the motion blur is due to camera translations only. Therefore, R_τ is a 3×3 identity matrix (I) and the homography simplifies to

$$H_\tau = K\left(I + \mathbf{t}_\tau \frac{\mathbf{n}^T}{d}\right) K^{-1}. \tag{1}$$

2.2 Normal from point-correspondences

The aim of our work is to use a single motion blurred image to estimate the surface normal of a planar scene. It is straightforward to show that the blur kernel centered at location $\mathbf{x}$ can be written as

$$h(\mathbf{x}, \mathbf{u}) = \frac{1}{E_t} \int_0^{E_t} \delta(\mathbf{u} - (H_\tau(\mathbf{x}) - \mathbf{x}))\, d\tau \tag{2}$$

i.e., the PSF represents the displacements undergone by an image point due to a set of motion transformations. The blur kernel induced will ideally consist of impulses at the corresponding shifts, and the weight of the impulse will be governed by the fraction of the exposure time spent in that homography/pose. For a fronto-parallel scene i.e., when $\mathbf{n} = [0\ 0\ 1]^T$, the blur induced would be space-invariant when camera undergoes only in-plane translations in the $xy-$plane. This is because for some transformation $\mathbf{t} = [T_{X_\tau}\ T_{Y_\tau}\ 0]^T$ and $\mathbf{n} = [0\ 0\ 1]^T$, we obtain

$$\begin{bmatrix} x_\tau \\ y_\tau \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{qT_{X_\tau}}{d} \\ 0 & 1 & \frac{qT_{Y_\tau}}{d} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3}$$

Clearly, the displacements in x and y directions are a constant (independent of the spatial location) and equal $\frac{qT_{X_\tau}}{d}$ and $\frac{qT_{Y_\tau}}{d}$, respectively. However, for a general inclined plane, the blur induced would be space-variant (due to change

in depth of the scene) even for pure in-plane translational motion. Corresponding to this situation, we will have (for $\mathbf{t} = [T_{X_\tau}\ T_{Y_\tau}\ 0]^T$)

$$\begin{bmatrix} x_\tau \\ y_\tau \\ 1 \end{bmatrix} = \begin{bmatrix} 1 + N_X \frac{T_{X_\tau}}{d} & N_Y \frac{T_{X_\tau}}{d} & qN_Z \frac{T_{X_\tau}}{d} \\ N_Y \frac{T_{Y_\tau}}{d} & 1 + N_Y \frac{T_{Y_\tau}}{d} & qN_Z \frac{T_{Y_\tau}}{d} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4}$$

Note that the displacements along x and y are no longer a constant and, in fact, vary as a function of the spatial location of the image point. Since our interest is in estimating the surface normal $\mathbf{n} = [N_X\ N_Y\ N_Z]^T$ (and not the camera motion per se), we rewrite equation (4) as

$$\begin{bmatrix} x_\tau \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 + N_X \frac{T_{X_\tau}}{d} \\ N_Y \frac{T_{X_\tau}}{d} \\ qN_Z \frac{T_{X_\tau}}{d} \end{bmatrix} \tag{5}$$

and

$$\begin{bmatrix} y_\tau \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} N_X \frac{T_{Y_\tau}}{d} \\ 1 + N_Y \frac{T_{Y_\tau}}{d} \\ qN_Z \frac{T_{Y_\tau}}{d} \end{bmatrix} \tag{6}$$

In equations (5) and (6), assuming point correspondences between $(x,\ y)$ and $(x_\tau,\ y_\tau)$ to be known, the unknowns are $N_X,\ N_Y,\ N_Z,\ T_{X_\tau},\ T_{Y_\tau}$ and d, and these appear in the right-most column vector. Note that the ratio $\frac{T_{X_\tau}}{d}$ (or $\frac{T_{Y_\tau}}{d}$) is a common scale factor multiplying the normal and hence need not be estimated. At first glance, it might appear that one can enforce unit norm on the normal to reduce the unknowns by one. However, we refrain from doing so since we lose the elegance of the linear equations (5) and (6) in the process. Thus, there are effectively three unknowns $(N_X,\ N_Y,\ N_Z)$ that are to be estimated. Hence, we need at least three point correspondences to solve this problem. If we can find point displacements at other locations in the image corresponding to the *same motion* $[T_{X_\tau}\ T_{Y_\tau}\ 0]^T$, then it should theoretically be possible to determine the unknowns. This, in fact, forms the basic premise for our method.

As discussed earlier in equation (2), the PSF or blur kernel encapsulates the displacements of pixels under the influence of camera motion. Thus, if we can establish point-correspondences (all influenced by the same motion) across atleast three blur kernels, then we can solve for the surface normal. However, because the blur kernel estimation itself is prone to small errors, it is only prudent that we use as many correspondences as possible. Note that we need to identify corresponding points among the PSFs with respect to the same *homography*. On this issue, we wish to

point out an interesting fact that a natural correspondence exists among the extremities of blur kernels (i.e., non-zero impulses at maximum distance from the origin of the PSF and on either side of the origin) across the image. We could potentially use the left (or right) extremity of the blur kernel in equation (5) or (6). Although it might appear that one can then solve for the normal, there is an ambiguity issue which we wish to highlight. Since the blur kernels are estimated independently across the image, there is a possibility of incurring spatial shifts in the PSFs when employing any blind deblurring method. A blurred patch b can be represented as convolution of latent patch l and blur kernel h i.e., $b = l * h$. Note that a shifted version (translational shift along x and y directions) of the true h also satisfies the convolution relation because $b(\mathbf{x}) = l(\mathbf{x} - \mathbf{s_0}) * h(\mathbf{x} + \mathbf{s_0})$. The shift introduced in the blur kernel is equivalently compensated in the latent image. Hence, if we choose only one extremity from the blur kernels, the surface normal cannot be estimated correctly due to possible misalignment errors. In order to resolve this issue, we choose the displacement *between the extremities* for computing correspondenes since this displacement is independent of any shift in the blur kernel.

From equation (5), the extremity of a PSF (say h_1) due to translation (say T_{X_p}) can be expressed as

$$\begin{bmatrix} x_{l_1} \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix} \begin{bmatrix} 1 + N_X \frac{T_{X_p}}{d} \\ N_Y \frac{T_{X_p}}{d} \\ qN_Z \frac{T_{X_p}}{d} \end{bmatrix} \tag{7}$$

where $(x_1\ y_1)$ is the spatial-location of the origin of h_1. Similarly, the $x-$coordinate of the right-extreme point of h_1 due to another translation (say T_{X_q}) will be

$$\begin{bmatrix} x_{r_1} \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix} \begin{bmatrix} 1 + N_X \frac{T_{X_q}}{d} \\ N_Y \frac{T_{X_q}}{d} \\ qN_Z \frac{T_{X_q}}{d} \end{bmatrix} \tag{8}$$

Subtracting equation (7) from equation (8), we get

$$\begin{bmatrix} x_{\triangle_1} \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix} \begin{bmatrix} N_X \frac{T_{X_q} - T_{X_p}}{d} \\ N_Y \frac{T_{X_q} - T_{X_p}}{d} \\ qN_Z \frac{T_{X_q} - T_{X_p}}{d} \end{bmatrix} \tag{9}$$

where $x_{\triangle_1}$ indicates the difference between the $x-$coordinates of the two extreme points of the blur kernel h_1. If we

can determine M such PSFs in the given blurred image, then we have a set of M (≥ 3) linear equations given by

$$\begin{bmatrix} x_{\triangle_1} \\ x_{\triangle_2} \\ . \\ . \\ x_{\triangle_M} \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ . & . & 1 \\ . & . & 1 \\ x_M & y_M & 1 \end{bmatrix} \begin{bmatrix} N_X \frac{T_{X_q} - T_{X_p}}{d} \\ N_Y \frac{T_{X_q} - T_{X_p}}{d} \\ qN_Z \frac{T_{X_q} - T_{X_p}}{d} \end{bmatrix} \tag{10}$$

where $(x_i\ y_i)$ represents the spatial-location of the origin of the i^{th} PSF. Note that $\frac{T_{X_q} - T_{X_p}}{d}$ is a constant that multiplies every component of $\mathbf{n}$ and hence need not be estimated. Therefore, one can solve equation (10) using least-squares to infer the surface normal.

The procedure explained above, in fact, is equally applicable to extreme points along the y direction too. Since our scheme relies on pixel motion, we propose to use $x_{\triangle_i}$ or $y_{\triangle_i}$, whichever is higher in magnitude. Note that the fronto-parallel plane is a special case of our formulation in that the PSFs will be identical at all locations i.e, $x_{\triangle_i} = k\ \forall i$ from which the solution can be inferred as $\mathbf{n} = [0\ 0\ 1]^T$.

Due to translational motion of the camera, the PSFs vary with the spatial location of the patch. A patch closer to the camera contains more blur as compared to a patch farther away from the camera. To determine the extremities of a PSF, we calculate the row sum and column sum of the PSF and choose the positions of the first and last non-zero values of the PSF as extreme points. These points are indicated by red (left-most) and green (right-most) pixels. Pixels with the same color constitute point correspondences. Therefore, all the red (green) points correspond to the same homography.

2.2.1 PSF estimation

Although our interest is not in estimating camera motion, we need to determine PSFs at different spatial locations in the blurred image. There exist several methods in the literature for blur kernel estimation. We used an off-the-shelf blind motion deblurring technique [7] to estimate the blur kernel for a selected patch. Estimating the PSF from a single motion blurred image is a very ill-posed problem since there exist many possible combinations of PSF and latent image that can lead to the same blurred image. Hence, blind motion deblurring methods typically impose priors on the PSF and the latent image. The method of [7] reveals that strong edges need not always lead to accurate PSF estimation and employs a two-phase approach to estimate PSF. In the first phase, the authors define a metric to identify

useful edges. These edges are considered to estimate a coarse blur kernel. In the second phase, an iterative support detection method is used (instead of hard-thresholding) to estimate the sparse blur kernel. The method executes fast and the accuracy of PSF estimation is also quite satisfactory [7].

2.3 Experiments

In this section, we validate the proposed method with examples, both synthetic and real. Since both PSF estimation as well as extreme point detection can involve small errors, we propose to use about 8 point-correspondences (instead of the minimum of 3) in equation (10) for robustness against noise. For the synthetic case, we choose focal length $q = 1200$ pixels which is a practical value. For these experiments, we assumed a surface normal, applied a set of homographies (camera translations) on an unblurred textured image, and computed the weighted average of the transformed images to yield the blurred observation. For the real case, the focal length (usually in mm) is gathered from the meta-data itself, and is converted into pixels using the sensor dimensions and the resolution of the image. The value of d in equation (10) is the same for all the points lying on the plane and it can be any constant (other than zero). In this work, we are interested only in the orientation of plane (and not in d which is embedded in the constant that multiplies $\mathbf{n}$ in equation (10)).

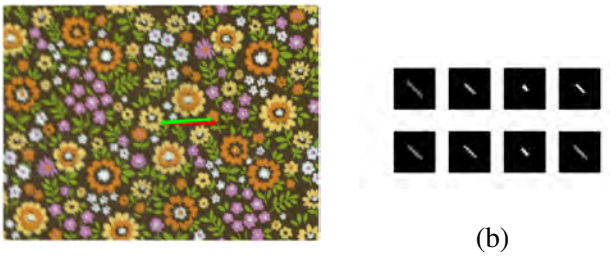
2.3.1 Synthetic case

In the first example, we assumed a fronto-parallel planar scene ($\mathbf{n} = [0\ 0\ 1]^T$). We applied a set of translations along both x and y directions and the blurred image thus obtained is shown in Fig. 1(a). Due to the fronto-parallel nature of the scene, all the 3D points are at the same distance from the camera and experience identical blur. We randomly select eight (spatially well-separated) patches in Fig. 1(a) and estimate their PSFs using [7]. These PSFs are shown in Fig. 1(b) and, as expected, have the same form. The extreme points in each PSF are detected as discussed earlier in section III and the displacements between these points is substituted in equation (10) to solve for the normal. The estimated normal turned out to be $\hat{\mathbf{n}} = [0\ 0.0632\ 0.998]$ which is quite close to the true normal.

Next, we use the same image as in the earlier example but assume an inclined plane with normal $\mathbf{n} = [-0.6428\ 0\ 0.7660]$. Following the procedure outlined earlier for the fronto-parallel case, a blurred observation (Fig. 2(a)) was generated using a set of transformations for the camera motion. We randomly selected eight patches (each of size 120×120 pixels) and the corresponding PSFs estimated using [7] are shown in Fig. 2(b). Note that the blur kernel is space-variant, as expected. The extreme point correpondences among the blur kernels have also been indicated in Fig.

(a) (b)

Figure 1: (a) A fronto-parallel scene with translational blur. (b) PSFs estimated using [7] at random locations in (a).



(a) (b)

Figure 2: (a) Inclined plane with motion blur. (b) PSFs estimated at different locations in (a).

4(b). From the displacements of the extremities, the normal was estimated using equation (10) by employing only the $x-$translations. The result was $\hat{\mathbf{n}} = [-0.6007\ 0.0370\ 0.7986]$ which is close to the actual normal. The angular error between the actual (red arrow) and the estimated normal (green arrow) is only 3.7 degrees as depicted in Fig. 2(a).

(a) (b)

Figure 3: (a) Fronto-parallel blurred image. (b) PSFs estimated at different locations in (a).

2.3.2 Real case

We used a Canon 60D camera to capture real data. The sensor width of the camera was 23.2 mm and the spatial resolution was 720×480 pixels. For the real experiments, we employed a translational stage to induce translational motion blur along both x and y directions.

In the first example, we captured a translationally blurred fronto-parallel textured board (Fig. 6(a)). Akin to the synthetic case, we choose eight different patches (again of size 120×120 pixels) and determine the PSFs corresponding to the center of these patches using [7]. The estimated PSFs are shown in Fig. 6(b) with extreme points marked. The normal estimated using equation (10) was found to be $\hat{\mathbf{n}} = [0\ 0\ 1]$. Since we know apriori that the scene is

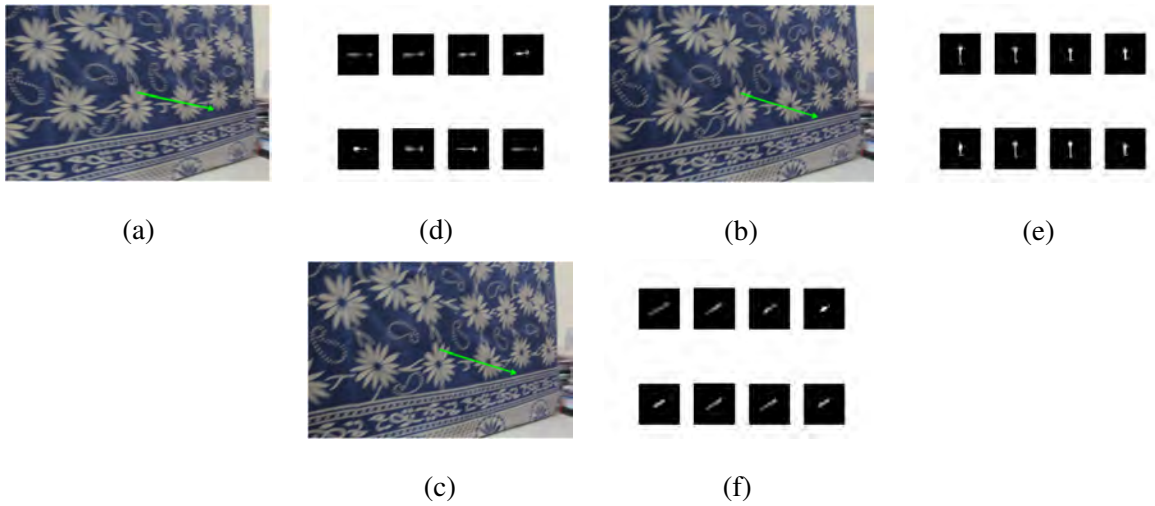


Figure 4: (a)-(c) Blurred images of an inclined plane for different camera translations. (d)-(f) PSFs corresponding to figures.(a)-(c), respectively.

fronto-parallel, we can conclude that the estimated normal is indeed correct.

Next, we captured a blurred image of an inclined plane as shown in Fig. 4(a). One can visually perceive the space-variant nature of the blur in this image. We randomly picked eight patches and the estimated PSFs are shown in Fig. 4(d). The extreme points in each PSF are represented with red (left-most) and green (right-most) colors. By following the same procedure discussed in the earlier experiments, the surface normal was found to be $\hat{\mathbf{n}} = [-0.4601\ 0.0970\ 0.8825]$. Since, this is a real example, we do not know the true normal. We ascertain the correctness of the estimated normal by capturing blurred images of the *same plane* but with two different camera translations. Ideally, the estimated normals should be identical irrespective of the camera motion. We captured two more blurred

images with different in-plane translations and these are shown in Figs. 4(b)-(c), with their corresponding PSFs (Figs. 4(e)-(f)). The estimated normals were found to be $\hat{\mathbf{n}}$= $[-0.4886 \ -0.051 \ 0.871]$ and $[-0.4601 \ 0.1400 \ 0.8767]$ respectively. Note that the estimated normals in all the three cases are quite close to one another reaffirming the correctness of our procedure. Furthermore, we physically measured the orientation of the plane and found it to be 30 degrees. This is indeed close to the value of 28 degrees obtained using the proposed method.

(a)

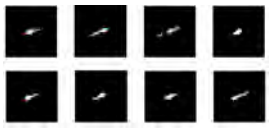
(b)

Figure 5: (a) Planar surface with motion blur. (b) PSFs estimated at different locations in (a).

We show another real example in Fig. 5(a). We captured an outdoor ground-plane with the optical axis of the camera approximately parallel to the ground-plane. The focal length was 18 mm. From Fig. 5(a), we observe that the image has significant variations in blur. The lower portion of the image (close to the camera) has more blur compared to the upper portion (far from the camera). To recover surface normal, we selected eight patches such that the patches were spreadout across the image. Their estimated PSFs are shown in Fig. 5(b). From the PSFs, we can infer that the translation is more prevalant along the $x-$direction. The detected extremities in each PSF are also indicated in Fig. 5(b). By following the procedure discussed earlier, the normal was computed as $[0.1373 \ 0.9800 \ 0.1442]$. Because the optical axis was not exactly parallel to the plane, the resultant angle turns out to be 81 degrees which is as expected.

In the final example, a planar scene was imaged as shown in Fig. 6(a). The bottom of the plane is closest to the camera while the top edge of the plane is the farthest. The focal length of the camera (18 mm) was obtained from the image meta-data. Using sensor dimensions and image size, the focal length translates to 581 pixels. We randomly picked eight patches throughout the image and their corresponding PSFs are shown in Fig. 6(b). After substituting the focal length and displacements of each PSF in equation (10), the computed surface normal was found to be $[0 \ -0.3713 \ 0.9151]$ and is indicated by a green arrow.

(a)



(b)

Figure 6: (a) A planar surface with translational motion blur. (b) Blur kernels extracted at different spatial locations in (a).

3 Efficient Change Detection for Large Motion Blurred Images

Feature-based approach is commonly used in image registration. There are several methods for feature extraction such as SIFT, SURF, ORB and MSER (Lowe et al. [8], Bay et al. [9], Rublee et al. [10], Matas et al. [11]). These algorithms are primarily designed to work on small to medium-sized images. Memory requirement is an important factor to consider when employing these approaches for high resolution images. Huo et al. [12] showed that SIFT features require a prohibitively huge amount of memory for very large images. Another drawback of feature-based approaches while working on large images is incorrect feature matching due to the occurrence of multiple instances of similar objects across the image (Carleer et al. [13]). Coarse-to-fine strategies for feature matching are followed by Yu et al. [14] and Huo et al. [12] to enable matching.

Within the scope of the problem tackled here, there is yet another deterrent in adopting feature-based approach and that is blur. Motion blur is a common occurrence in aerial imagery where the imaging vehicle is always on the move. In addition to geometric matching, photometric matching becomes essential in such a scenario. Feature-based approaches are not designed to handle the presence of blur and fail to reliably detect features in the presence of blur. A traditional approach to handle this situation is to first deblur the observation, and then pass on the resultant image to the change detection pipeline where it is compared with a clean reference image after feature-based registration. A number of approaches already exist in the literature to perform deblurring. Blind deconvolution methods recover a sharp image from the blurred image with an unknown blur kernel under the assumption of space-invariant blur. Fergus et al. [15] take a natural image statistics based Bayesian approach to estimate the blur kernel and deblur using Richardson-Lucy algorithm. A two-phase approach with kernel initialisation using edge priors and kernel refinement

based on iterative support detection is employed by Xu et al. [16] for kernel estimation, and the deblurring is sought through TV-ℓ_1 deconvolution. Space-variant blur based approaches include that of Gupta et al. [17] who model a motion density function to represent the time spent in each camera pose and to generate spatially varying blur kernels and eventually restore the deblurred image using a gradient-based optimisation. Whyte et al. [18] define a transformation spread function for space-variant blur similar to the point spread function for space-invariant blur to restore the motion blurred image using MAP approach. Hu et al. [19] estimate weights for each camera pose in a restricted pose space using a backprojection model while deblurring is carried out by employing a gradient-based prior. Leveraging gradient sparsity, Xu et al. [20] proposed a unified framework to perform both uniform and non-uniform image deblurring.

An issue with such a deblur-difference framework is that it must deal with the annoying problem of artifacts that tend to get introduced during the course of deblurring. A more serious issue within the context of this work is that none of the deblurring methods are designed to handle very large images. Furthermore, the deblurring methods would fail if the occluder was not static since the image will then be governed by two independent motions.

In the problem of change detection, the goal is to detect the difference between a reference image with no artifacts and an observed image which is blurred and has viewpoint changes as well. We develop a unified framework to register the reference image with the blurred image and also to detect occlusions simultaneously. The occluder is not constrained to be static. To address the issue of image size, we reveal that the camera motion can be elegantly extracted from only a part of the observation. For reasons discussed earlier, we follow a reblur-difference pipeline instead of a deblur-difference pipeline. While Punnappurath et al. [21] also followed a reblur-difference strategy, our work is more general and, in fact, subsumes their work. Specifically, we use an optimisation framework with partial non-negative constraint which can handle occlusions of any polarity, and we efficiently tackle the issue of large image dimension. In addition, our algorithm can also deal with dynamic occluders. In our approach, the estimated camera motion is used to reblur the reference image to photometrically match it with the observed image, and thereby detecting the changes.

We develop a scheme to automatically select good sub-images from the given observation to enable reliable estimation of the camera motion. We propose a memory and computationally efficient registration scheme to estimate the camera motion from the selected sub-image, irrespective of the presence or absence of occlusions in the sub-image. We advocate a reblur-difference pipeline for geometric as well as photometric registration of the reference image and the blurred observation for robust change detection.

3.1 Blur, Registration and Occlusion

In this section, we briefly discuss motion blur model in a camera. We then show how to invoke an optimisation framework to simultaneously register the reference image with the blurred image as well as detect occlusions, if any.

3.1.1 Motion Blur Model

Each pixel in a digital camera embeds a sensor which collects photons from the scene. A digital circuit provides the intensity value based on the number of photons received. All the pixels are exposed for a finite amount of period T_e which is the exposure time of the image. The resultant intensity at each pixel is the average of all intensities that the pixel sees during the exposure period. Let us denote the camera path during the image exposure period by $\mathbf{p}(t)$ for $0 \leq t \leq T_e$. Let $\mathbf{f}$ represent the image observed by the camera during an infinitesimal amount of time. Let $\mathbf{g}$ be the image observed by the camera with an exposure time T_e. Let the number of rows and columns in the images be M and N respectively, so that $\mathbf{f}, \mathbf{g} \in \mathbb{R}^{MN \times 1}$. Then, we have

$$\mathbf{g} = \frac{1}{T_e} \int_0^{T_e} \mathbf{f}_{\mathbf{p}(t)} \, dt. \tag{11}$$

where $\mathbf{f}_{\mathbf{p}(t)}$ is the image observed by the camera due to the pose $\mathbf{p}(t)$ at a particular time t.

When there is no motion, the camera observes the same scene during the entire exposure time, and hence a clean image without any blur is observed. In this case, $\mathbf{p}(t) = \mathbf{0}$ for all $0 \leq t \leq T_e$, and $\mathbf{g} = \mathbf{f}$. Thus $\mathbf{f}$ represents also the image seen by the camera with no motion during the exposure time T_e. In the presence of camera motion, the sensor array records different scenes at every instant during the exposure time. The resultant image thus embodies blur in it, and we have $\mathbf{g} \neq \mathbf{f}$.

We discretise the continuous model in (11) with respect to a finite camera pose space $\mathcal{P}$. We assume that the camera can undergo only a finite set of poses during the exposure time. Let us define $\mathcal{P} = \{\mathbf{p}\}_{i=1}^{|\mathcal{P}|}$ as the set of possible camera poses. We can write (11) equivalently as

$$\mathbf{g} = \sum_{\mathbf{p}_k \in \mathcal{P}} \omega_{\mathbf{p}_k} \, \mathbf{f}_{\mathbf{p}_k} \tag{12}$$

where $\mathbf{f}_{\mathbf{p}_k}$ is the warped reference image $\mathbf{f}$ due to the camera pose $\mathbf{p}_k$. Each scalar $\omega_{\mathbf{p}_k}$ represents the fraction of exposure time that the camera stayed in the pose $\mathbf{p}_k$. Thus we have $\sum_{\mathbf{p}_k} \omega_{\mathbf{p}_k} = 1$ if the camera takes only the poses from the defined pose set $\mathcal{P}$. The weights of all poses are stacked in the pose weight vector $\boldsymbol{\omega}$. Since the averaging

effect removes the time dependency of the continuous camera path $\mathbf{p}(t)$, this discretisation model is valid. We assume that the scene is far enough from the camera such that planarity can be assumed.

3.1.2 Joint Registration and Occlusion Detection

We now consider the problem of estimation of camera poses during exposure. Given a reference image $\mathbf{f}$ which is captured with no camera motion, and a blurred image $\mathbf{g}$ arising from an unknown camera motion, the following problem can be posed to solve for the camera motion.

$$\widetilde{\boldsymbol{\omega}} = \min \|\mathbf{g} - \mathbf{F}\boldsymbol{\omega}\|_2^2 + \lambda\|\boldsymbol{\omega}\|_1 \text{ subject to } \boldsymbol{\omega} \succeq 0 \tag{13}$$

Here $\mathbf{F}$ is the matrix which contains the warped copies of the reference image $\mathbf{f}$ in its columns for the camera poses in $\mathcal{P}$. In the whole pose space, the camera can be moved through only a small set of poses. This is prioritised as the ℓ_1 norm in (13) which promotes sparsity of the pose weight vector. The above problem seeks the sparsest non-negative pose weight vector which satisfies the relation between reference and blurred images. Matrix-vector multiplication $\mathbf{F}\boldsymbol{\omega}$ is an equivalent form of (12).

This model, however, does not accommodate occluding objects in the observation $\mathbf{g}$ although this is quite often the case in aerial surveillance systems. To handle this, let $\mathbf{g}_{\text{occ}}$ be the observed image captured with blur and occlusions. We model the occlusion as an additive term to $\mathbf{g}$ to give $\mathbf{g}_{\text{occ}} = \mathbf{g} + \boldsymbol{\chi}$. The occluded image $\boldsymbol{\chi}$ can take both positive and negative values since the occluded pixels can have intensities greater or lesser than the intensities purely explained by blur. This model can then be written as

$$\mathbf{g}_{\text{occ}} = \begin{bmatrix} \mathbf{F} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{\chi} \end{bmatrix} = \mathbf{A}\boldsymbol{\xi}. \tag{14}$$

Here $\mathbf{A}$ is a combined dictionary of warped reference images to represent blur and the $N \times N$ identity matrix to represent occlusions, where $\boldsymbol{\xi}$ is the combined weight vector, the first $|\mathcal{P}|$ elements of which represent the pose weight $\boldsymbol{\omega}$ and the remaining N elements represent the occlusion vector $\boldsymbol{\chi}$. To solve this under-determined system, we leverage the prior information about the camera motion and occlusion, viz. the sparsity of the camera motion in the pose space and the sparsity of the occlusion in the spatial domain. Thus we impose ℓ_1 norm prior on $\boldsymbol{\xi}$. We estimate the combined weight vector by solving the following optimisation problem.

$$\widetilde{\boldsymbol{\xi}} = \arg\min_{\boldsymbol{\xi}} \|\mathbf{g}_{\text{occ}} - \mathbf{A}\boldsymbol{\xi}\|_2^2 + \lambda\|\boldsymbol{\xi}\|_1 \text{ subject to } \mathbf{C}\boldsymbol{\xi} \succeq 0 \tag{15}$$

where $\mathbf{C} = \begin{bmatrix} \mathbf{I}_{|\mathcal{P}|} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$. As mentioned earlier, the occlusion vector can take both positive and negative values. Thus, unlike the work of Punnappurath et al. [21] who modify the signs of the identity matrix, we neatly impose non-negativity constraint *only* on the elements of the pose weight vector.

3.2 Registration of Very Large Images

Building the matrix $\mathbf{A}$ in (14) is a crucial step in our problem. The occlusion part of the matrix $\mathbf{I}_N$ can be stored and processed efficiently since it is a diagonal matrix. The first part of the matrix $\mathbf{F}$ contains the warped versions of $\mathbf{f}$ for all the poses in $\mathcal{P}$. Though the reference image $\mathbf{f}$ operates in the intensity range [0-255] and requires only an unsigned 8-bit integer for each pixel, this is not the case for the storage of the warped versions. The pixel values of the warped image $\mathbf{f}_{\mathbf{p}_k}$ can take floating-point values due to bilinear interpolation during its generation. A round-off during the interpolation makes the equality in (12) only approximate, and hence it might lead to a wrong solution. A single warped image needs MNd bits of storage memory for operation, where d is the number of bits required to store a floating-point number. For even a 25 mega-pixel image with 5000 rows and 5000 columns and with $d = 32$ bits, a warped image requires $5000 \times 5000 \times 32$ bits, that is 95.3 megabytes. If all three colour channels are used, this value will triple. Storing all warps for the pose space as the matrix $\mathbf{F}$ thus warrants a huge amount of memory allocation which is infeasible in practical situations.

3.2.1 Pose Weight Estimation from Sub-images

Our solution to the large image problem stems from the interesting observation that all the pixels in an image observe the same camera motion during the exposure period. We leverage this fact to estimate the pose weight vector from a *subset* of pixels in the image. Let $\mathbf{f}^{(S)}$ and $\mathbf{g}^{(S)}$ represent a portion of the reference and blurred images, respectively. The sub-image size is $S \times S$, and $\mathbf{f}^{(S)}, \mathbf{g}^{(S)} \in \mathbb{R}^{S^2 \times 1}$. We call these, respectively, as reference sub-image and blurred sub-image. We will ignore the presence of occlusion in this discussion for clarity. The relation in (12) holds for $\mathbf{f}^{(S)}$ and $\mathbf{g}^{(S)}$ as well i.e.

$$\mathbf{g}^{(S)} = \sum_{\mathbf{p}_k \in \mathcal{P}} \omega_{\mathbf{p}_k} \, \mathbf{f}^{(S)}_{\mathbf{p}_k} \tag{16}$$

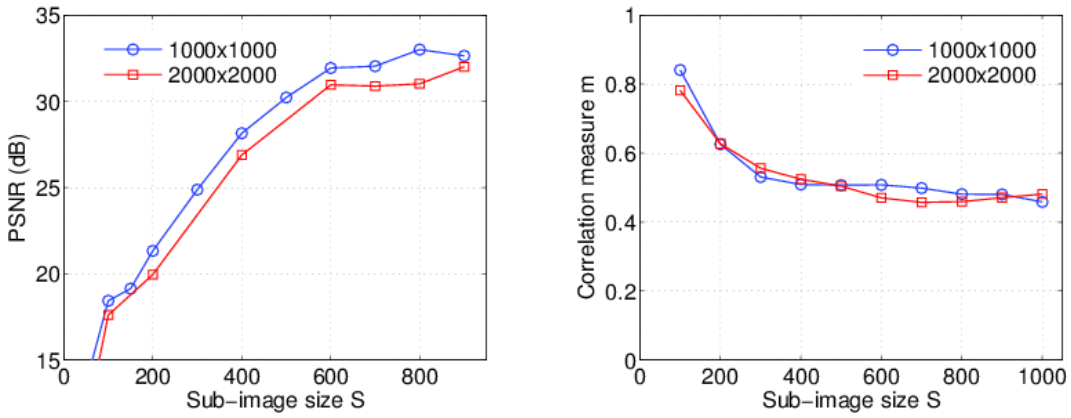
The estimated pose weight vector $\boldsymbol{\omega}$ will be the same irrespective of whether we use $\mathbf{f}$ and $\mathbf{g}$ or $\mathbf{f}^{(S)}$ and $\mathbf{g}^{(S)}$ in (13). We propose to estimate the camera motion using only the sub-images thus effectively circumventing the issue

Figure 7: Shown are some of the reference (top row) and blurred (bottom row) images used in our experiments in Section 3.2.

of memory storage.

To verify our proposition, we now perform experiments to estimate camera motion from sub-images of large synthetically blurred images. We simulate five different continuous camera paths for a predefined set of discrete translation and rotation ranges. We use a set of five images for this experiment. We thus have a set of five reference images $\mathbf{f}$ and 25 blurred images $\mathbf{g}$. Some of the reference and blurred images are shown in Fig. 7. We pick a pair of $\mathbf{f}$ and $\mathbf{g}$, and for a given S we pick the sub-images $\mathbf{f}^{(S)}$ and $\mathbf{g}^{(S)}$. Using these two images, we estimate the pose weight vector $\boldsymbol{\omega}$ using (13). Since the motion involves combinations of rotations and translations, direct comparison of original and estimated motion vectors may not lead to a correct measure of error. Hence we measure the success of our estimation by reblurring. We warp $\mathbf{f}$ using the poses in $\mathcal{P}$ with the estimated weights $\widetilde{\boldsymbol{\omega}}$, and perform a weighted average of the warps, resulting in a reblurred reference image $\mathbf{f}$. We then calculate the reconstruction PSNR of the reblurred reference image with respect to the original blurred image $\mathbf{g}$. If the motion estimation from the sub-image is correct, then the reblurred image will be close in appearance to the original blurred image resulting in a high PSNR. We repeat this experiment for different values of S. The variation of PSNR with respect to S is shown in Fig. 8(a) for image sizes of 1000×1000 and 2000×2000.

For small values of S, the variation of motion blur within the sub-image will be small and will approximately tend to mimic space-invariant blur. Hence solving (13) results in a wrong pose weight estimate which results in a poor PSNR between the reblurred and blurred images. The PSNR increases as S increases since the blur variation inside the sub-image also increases. We observe that the PSNR value stabilises after a particular value of S. Beyond this



(a) (b)

Figure 8: (a) PSNR in dB, and (b) correlation measure (for different sub-image sizes S). Original image sizes are 1000×1000 (blue circle) and 2000×2000 (red square).

$S = 100$ $S = 300$ $S = 600$

Figure 9: Estimated blur kernels for different sub-image sizes S. The blur kernels are displayed as binary images with non-zero values shown in white colour.

point, any further increase in S results only in marginal benefits in terms of correct estimation of pose weights. The size of the sub-image is an important factor in estimating the true camera motion. Too small an S renders the notion of space-variant blur inside the sub-image invalid, and results in a wrong pose weight estimate. Too large an S will kindle storage and processing problems. In the following subsection, we formulate a method to automatically choose good sub-images for reliably estimating the camera motion.

3.2.2 Choosing a Good Sub-image

It is important to devise an automatic method to select a sub-image of a particular size at a particular location from the given large blurred observation. We develop a measure that would indicate the quality of the selected sub-image for estimating the camera motion. Given a pair of reference and blurred sub-images $\mathbf{f}^{(S)}$ and $\mathbf{g}^{(S)}$ of size S, we randomly select N_h scattered locations across the image. We crop small patches, $\mathbf{f}_k^{(S)}$ and $\mathbf{g}_k^{(S)}$, from $\mathbf{f}^{(S)}$ and $\mathbf{g}^{(S)}$ respectively, for $k = 1$ to N_h. We approximate the blur to be space-invariant in these patches, and estimate blur kernels using (13) allowing the pose space to contain only in-plane translations. Let us denote these blur kernels by $\mathbf{h}_k$ for $k = 1$ to N_h.

If the selected sub-image has sufficient variations in blur across it, then each of these blur kernels will be different as they are quite spread out spatially. Hence a comparison of these estimated kernels is a good way to decide the suitability of the sub-image for motion estimation. We advocate the use of normalised cross-correlation of the kernels for this decision. Normalised cross-correlation between two 2D kernels $\mathbf{h}_i$ and $\mathbf{h}_j$ is given by

$$\mathbf{NCC}(\mathbf{h}_i, \mathbf{h}_j) = \frac{\mathbf{corr}(\mathbf{h}_i, \mathbf{h}_j)}{\|\mathbf{h}_i\|_2 \|\mathbf{h}_j\|_2}. \tag{17}$$

Values of the matrix $\mathbf{NCC}$ lie in $[0, 1]$. We use the maximum value of this matrix as our measure to compare the blur kernels, i.e.,

$$\text{Correlation measure } m(\mathbf{h}_i, \mathbf{h}_j) = \max\ \mathbf{NCC}(\mathbf{h}_i, \mathbf{h}_j) \tag{18}$$

Note that $m(\mathbf{h}_i, \mathbf{h}_j)$ attains a peak value of 1 if the two blur kernels are same. If the sub-image size is small, then there will not be sufficient blur variations across it, and our measure value will be close to 1. If the kernels are dissimilar, then m takes values close to 0.

Fig. 9 shows four blur kernels of the patches that are extracted randomly from sub-images of sizes $S = 100, 300$ and 600. The patch size used is 41×41 and $N_h = 4$. Blur kernels corresponding to space-invariant blur will appear the same irrespective of the spatial point. For a small sub-image of size $S = 100$, it can be clearly observed that the four kernels are similar. Hence the camera motion cannot be correctly explained by this sub-image. For $S = 300$, the blur kernels are more dissimilar, and for $S = 600$, they look completely different. Thus, higher values of S describe the motion better. From these four blur kernels, six measure values m are estimated for every pair. Fig. 8 (b) shows the plot of mean $\overline{m}$ of these six values with respect to the sub-image size. The curve falls with increase in sub-image size as expected due to continuous decrease in kernel similarity. A synonymity can be observed between the plots in Figs. 8 (a) and (b). Correlation measure decreases initially with increasing S and stays almost constant after a certain

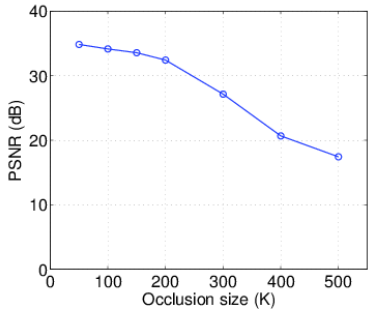


Figure 10: (a) PSNR in dB for $S = 600$ and different occlusion sizes K.

value of S. Similarly, the reconstruction PSNR stabilises after it reaches a particular sub-image size. Based on these observations, we define a threshold $T_m = 0.6\,\overline{m}_{100}$, where $\overline{m}_{100}$ is the correlation measure for $S = 100$, to accept or reject a sub-image for motion estimation. If $\overline{m}_{S_0}$ for a sub-image of a specific size S_0 is less than this threshold, we decide that the quality of the selected sub-image of size S_0 is good, and that the camera motion can be estimated from it.

3.2.3 Presence of Occlusion

A natural question to ask is how well our algorithm fares when there is occlusion in the selected sub-image itself. We add a random occlusion patch of size $K \times K$ to the reference image $\mathbf{f}$. We blur this image using the generated camera motion path, the resultant image being $\mathbf{g}_{\text{occ}}$. We slice the sub-images $\mathbf{f}^{(S)}$ and $\mathbf{g}_{\text{occ}}^{(S)}$, from $\mathbf{f}$ and $\mathbf{g}_{\text{occ}}$ respectively. We do not restrict the position of the sub-image with respect to the occlusion. Therefore, the sub-image can either include the full occlusion or a part of the occlusion or be devoid of the occlusion completely. Our combined dictionary $\mathbf{A}$ in (14) tackles both the presence of blur and occlusion simultaneously. If occlusion is present either fully or partially, it would be accommodated by the weights of the identity matrix in $\mathbf{A}$. If there is no occlusion present, then the occlusion weight vector will be zero. Thus, irrespective of the absence or presence (complete or partial) of the occluder in the sub-image, our formulation can elegantly handle it.

We next discuss the effect of the size of the occlusion for a chosen sub-image size S. We consider the worst case of the occlusion being present completely inside the chosen sub-image. We solve the optimisation problem in (15) with $\mathbf{f}^{(S)}$ and $\mathbf{g}_{\text{occ}}^{(S)}$ to arrive at the combined pose weight and occlusion weight vectors. Using the estimated $\widetilde{\boldsymbol{\omega}}$, we reblur the large reference image $\mathbf{f}$. We compare this reblurred image with the large blurred image $\mathbf{g}_{\text{occ}}$ ignoring the values in the occlusion region, since this comparison is to verify the success of our motion estimation. Fig. 10 shows

how the PSNR varies with respect to the value of K for $S = 600$. We note that our algorithm tackles the presence of occlusion quite well. The motion estimation is correct and thus PSNR values are good even when the occluder occupies up to half the sub-image area.

Algorithm 1 shows our complete framework of choosing good sub-images automatically, estimation of motion and change detection. In our experiments, we use an α value of 5 and the upper limit of S, S_{max}, is chosen as 900 based on the many experiments we carried out.

Algorithm 1:

Inputs: Reference image $\mathbf{f}$, blurred and occluded image $\mathbf{g}$.

Init: Pick four sub-images $\mathbf{f}^{(100)}$ based on Hu et al. [22], extract blur kernels and calculate $\overline{m}$ for each of the kernels. Average the four values to get $\overline{m}_{100}$.

Let $S = 200$.

1. Pick a sub-image of size S. If $\overline{m} < 0.6\,\overline{m}_{100}$, goto Step 5. Else, choose a different sub-image of the same size at a different location.
2. If a particular S is chosen α times, update $S \leftarrow S + 100$. Goto Step 2.
3. If $S > S_{\text{max}}$, declare blur to be space-invariant. Use one of the estimated blur kernels itself as the camera pose weight vector. Goto Step 6.
4. Estimate pose weight vector and occlusion weight vector for the selected sub-images $\mathbf{f}^{(S)}$ and $\mathbf{g}^{(S)}$ using (15).
5. Reblur the original reference image $\mathbf{f}$ using the estimated pose weight vector $\widetilde{\boldsymbol{\omega}}$.
6. Detect the changes by differencing the reblurred image and the original blurred image $\mathbf{g}$.

3.3 Experiments

We first evaluate the performance of our algorithm using a synthetic example. A reference image of size 2188×1315 pixels is shown in Fig. 11(a). We selected the pose space as follows- in-plane translations: $[-8{:}1{:}8]$ pixels, in-plane rotations: $[-3\circ : 1\circ : 3\circ]$. These ranges are also practically meaningful. To simulate blur incurred due to camera shake, we manually generated camera motion with a connected path in the pose space and initialized the weights. The synthesized camera motion was then applied on the same scene taken from a different view point with synthetically added occluders to produce the blurred and occluded image in Fig. 11(b).

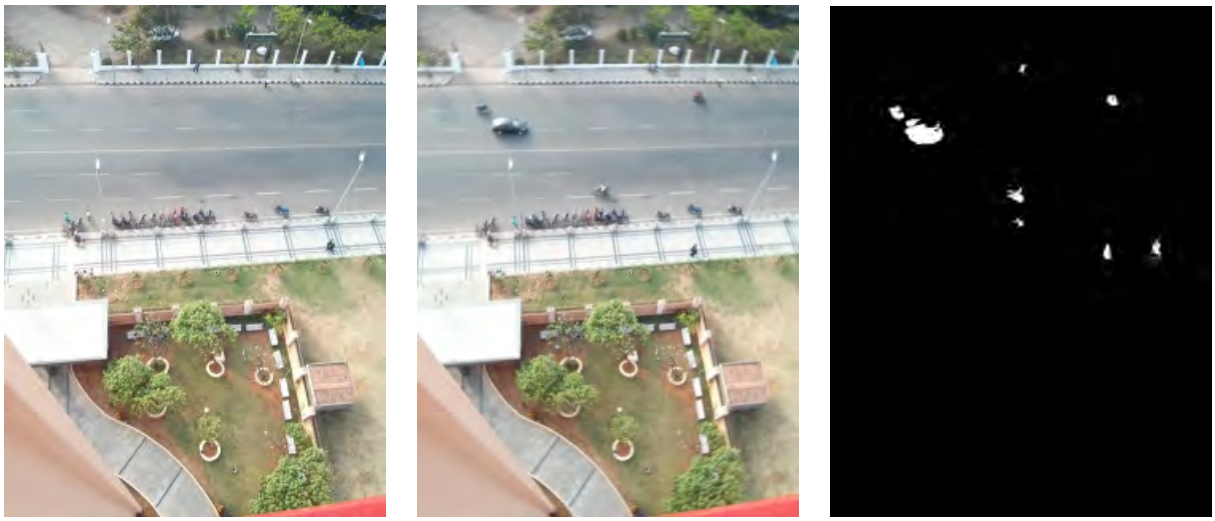
To evaluate the proposed method, we followed the steps outlined in Algorithm 1, selected four sub-images (based on Hu et al. [22]) of size 100×100 pixels, and calculated $\overline{m}_{100}$ independently for each. The average value of $\overline{m}_{100}$ was computed. Next, we picked a sub-image of size 200×200 pixels and calculated $\overline{m}$. The four kernels computed within the sub-image bore a large degree of similarity indicating that the space-varying nature of the blur was not being captured at this size. This step was repeated for five different sub-images of size 200×200 pixels but the value of $\overline{m}$ they yielded was approximately equal to $\overline{m}_{100}$ revealing that only a bigger sized sub-image can encapsulate the space-varying camera motion. Our algorithm converged for a sub-image of size 500×500 pixels where the computed $\overline{m}$ was less than $0.6\ \overline{m}_{100}$. The selected sub-images of size 500×500 pixels from the focused, and the blurred and occluded input images are shown in Figs. 11(c) and 11(d), respectively. The position of these sub-images have been indicated by red boxes in Figs. 11(a) and (b). Note that the selected sub-image, incidentally, does not contain any occlusion. To handle pose changes between the two images, we first coarsely aligned the reference image and the blurred and occluded image at a lower resolution using a multiscale implementation similar to [21]. Fig. 11(e) shows the reference image reblurred using the estimated $\widetilde{\boldsymbol{\omega}}$. The detected occlusions shown in Fig. 11(f) are found by comparing the blurred and occluded observation (Fig. 11(b)) with the reblurred reference image (Fig. 11(e)).

For our next experiment, we use the publicly available VIRAT database (Oh et al. [23]) which is a benchmark for video surveillance and change detection. Two frames corresponding to the reference image and the occluded image (shown in Figs. 12(a) and (b), respectively) were manually extracted from an aerial video. The frames are at the resolution of the original video i.e., 720×480 pixels. Since the resolution is low, we run our algorithm directly on the whole image instead of a sub-image. The detected occlusion is shown in Fig. 12(c). Although, strictly speaking, the images are not high resolution at all, the purpose is to demonstrate the efficiency of our method for aerial imaging. Also, this example illustrates how the proposed method elegantly subsumes the work in [21] for the case of low

Figure 11: (a) Reference image, (b) synthetically blurred and occluded observation from a different view point, (c) sub-image from (a), (d) sub-image from (b), (e) reference image reblurred using the estimated camera motion, and (f) detected occlusion.

(a) (b) (c)

Figure 12: (a) Reference image, (b) real blurred and occluded observation, and (c) detected occlusion.



(a) (b) (c)

Figure 13: (a) Reference image, (b) real blurred and occluded observation, and (c) multiple occluders found.

resolution images.

A final real example is shown in Fig. 13. The two images in Figs. 13 (a) and (b) were captured from the same view point but with a small time lapse using a Google Nexus 4 mobile phone which has an 8 MP camera. Observe how even small occluders with intensities close to the background are correctly detected by our algorithm (Fig. 13(c)). This example threw up small spurious non-connected occlusions in the bottom half of the image due to movement of leaves, and these were removed by simple post-processing.

To perform quantitative assessment of our method, we computed the following metrics which are well known in the change detection community: percentage of correct classification (PCC), Jaccard coefficient (JC) and Yule coef-

ficient (YC) [24]. For the real experiments, the ground-truth occlusion was assessed by asking different individuals to locally mark out the occluded regions as per their individual perception. The efficacy of our algorithm is further evident from the values in Table 1.

Table 1: Quantitative metrics for our results in Figs. 11 to 13.

Fig.	PCC	JC	YC
11	99.5957	0.7792	0.9348
12	99.7195	0.6218	0.9211
13	99.3736	0.4557	0.8729

The maximum size of the images considered was 18 MP. Despite our best attempts, we could not find any database containing image pairs of very large sizes (of the order of 100M) that could be used for testing. Nevertheless, the framework proposed here has the potential to handle even very large images. Due to file size constraints, we have included only the downscaled images in the pdf, and not the original high resolution images.

4 Space-variant Deblurring of Aerial Imagery

Blur in images resulting from motion of the camera during exposure time is an issue in many areas of optical imaging such as remote sensing, aerial reconnaissance and digital photography. For instance, the images captured by cameras attached to airplanes or helicopters are blurred due to both the forward motion of the aircraft, and vibrations. Manufacturers of aerial imaging systems employ compensation mechanisms such as gyroscope gimbals to mitigate the effect of vibrations. Although this reduces the blur due to jitter to some extent, there is no straightforward way to do the same for the forward movement. Moreover, these hardware solutions come at the expense of higher cost, weight and energy consumption. A system that can remove the blur by algorithmic post-processing provides an elegant solution to this problem.

Traditionally, image restoration techniques have modeled blurring due to camera shake as a convolution with a single blur kernel [25, 26, 27, 28, 29]. However, it is a well-established fact that the convolution model that employs a uniform blur kernel or point spread function (PSF) across the image is not sufficient to model the blurring phenomenon if the motion is not composed merely of in-plane translations. In fact, camera tilts and rotations occur frequently [30] and the blur induced by camera shake is typically non-uniform. This is especially true in the case of aerial imagery

where the blur incurred is not just due to the linear motion of the aircraft but also due to vibrations. Approaches to handling non-uniform blur broadly fall into two categories. The first relies on local uniformity of the blur. Based on the assumption that a continuously varying blur can be approximated by a spatially varying combination of localized uniform blurs, Hirsch et al. [31] propose a method to restore non-uniform motion blur by using an efficient filter flow framework. Building on the idea of a motion density function, yet another scheme for space-varying blur has been proposed by Gupta et al. [32]. The motion-blurred image is modeled by considering the camera motion to be comprised only of in-plane translations and in-plane rotations. The second and more recent non-uniform deblurring approach uses an elegant global model [30, 33, 34] in which the blurred image is represented as the weighted average of warped instances of the latent image, for a constant depth scene. The warped instances can be viewed as the intermediate images observed by the camera during the exposure time when the camera suffers a shake. Tai et al. [35] have proposed a non-blind deblurring scheme based on modifying the Richardson Lucy deconvolution technique for space-variant blur. However, they assume that the blurring function is known *a priori* and does not need to be estimated. Whyte et al. [30, 36] propose a non-uniform image restoration technique where the blurring function is represented on a 3D grid corresponding to the three directions of camera rotations. As pointed out in [34] the main disadvantage of this global geometric model is heavy computational load due to the dense sampling of poses in the high dimensional camera motion space. A common approach to tackle this problem is to adopt a multi-scale approach that involves constructing an image pyramid and using coarse-grained sampling. But this simplification inevitably introduces reconstruction errors [34]. Hu and Yang [34] present a fast non-uniform deblurring technique by using locally estimated blur kernels to restrain the possible camera poses to a low-dimensional subspace. But the kernels themselves need to be input by the user and the final deblurring quality is dependent on the accuracy of the estimated PSFs. An unnatural L_0 sparse representation for uniform and non-uniform deblurring has also been proposed recently by Xu et al. [37]. In the hardware-assisted restoration techniques, Joshi et al. [38] attach sensors to the camera to determine the blurring function, while Tai et al. [39] propose a deblurring scheme that uses coded exposure and some simple user interactions to determine the PSF.

In this work, we propose a fully blind single image non-uniform deblurring algorithm suited for aerial imagery that does not require any additional hardware. We reduce computational overhead by approximating the camera motion with a 3D pose space and optimizing only over a subspace of 'active' camera poses. This reduction in dimensionality allows us to use dense sampling and our results compare favourably with state-of-the-art deblurring algorithms. In contrast to [34], our alternating minimization algorithm, which uses a novel camera pose initialization and pose

perturbation step, works on the global geometric model and doesn't require the calculation of blur kernels at various image locations, thereby eliminating the need for user interaction.

4.1 The motion blur model

In this section, we review the non-uniform blur model for aerial images. Since the distances involved are quite large, the ground scene can be modeled as being approximately planar. When the motion of the camera is not restricted to in-plane translations, the paths traced by scene points in the image plane will vary across the image resulting in space-variant blur. The convolution model with a single blur kernel does not hold in such a scenario. However, the blurred image can be accurately modeled as the weighted average of warped instances of the latent image using the projective model in [40, 30, 32, 34], when the scene is planar. In the discrete domain, this can be represented as

$$b\left(i,j\right)=\sum_{k\in\mathbf{T}}\omega\left(k\right)l\left(\mathcal{H}_{k}\left(i,j\right)\right) \tag{19}$$

where $l(i,j)$ denotes the latent image of the scene, $b(i,j)$ is the blurred observation, and $\mathcal{H}_k(i,j)$ denotes the image coordinates when a homography $\mathcal{H}_k$ is applied to the point (i,j). The parameter ω, also called the *transformation spread function* (TSF) [40] in the literature, depicts the camera motion, and $\omega(k)$ denotes the fraction of the total exposure duration for which the camera stayed in the position that caused the transformation $\mathcal{H}_k$. Akin to a PSF, $\sum_{k\in\mathbf{T}}\omega\left(k\right)=1$. The TSF ω is defined on the discrete transformation space $\mathbf{T}$ which is the set of sampled camera poses. The transformation space is discretized in such a manner that the difference in the displacements of a point light source due to two different transformations from the discrete set $\mathbf{T}$ is at least one pixel. Note that although the apparent motion of scene points in the image will vary at different locations when the camera motion is unrestricted, the blurring operation can still be described by a single TSF using equation (19). For example, if the camera undergoes only in-plane rotations, the TSF will have non-zero weights only for the rotational transformations. Observe that if the camera motion is confined to 2D translations, the PSF and TSF will be equivalent.

If $\mathbf{l}$, $\mathbf{b}$ represent the latent image and the blurred image, respectively, lexicographically ordered as vectors, then, in matrix-vector notation, equation (19) can be expressed as

$$\mathbf{b}=\mathbf{A}\boldsymbol{\omega} \tag{20}$$

where $\mathbf{A}$ is the matrix whose columns contain projectively transformed copies of $\mathbf{l}$, and $\boldsymbol{\omega}$ denotes the vector of weights $\omega(k)$. Note that $\boldsymbol{\omega}$ is a sparse vector since the blur is typically due to incidental camera shake and only a small fraction

of the poses in $\mathbf{T}$ will have non-zero weights in $\boldsymbol{\omega}$. Alternately, $\mathbf{b}$ can also be represented as

$$\mathbf{b} = \left(\sum_{k \in \mathbf{T}} \omega(k) \mathbf{H_k} \right) \mathbf{l} = \mathbf{B}\mathbf{l} \tag{21}$$

where $\mathbf{H_k}$ is the matrix that warps the latent image $\mathbf{l}$ according to the homography $\mathcal{H}_k$, while $\mathbf{B} = \sum_{k \in \mathbf{T}} \omega(k) \mathbf{H_k}$ is the matrix that performs the non-uniform blurring operation. Note that $\mathbf{B}$ is a sparse square matrix that can be efficiently stored in memory and each row of $\mathbf{B}$ corresponds to the blur kernel at that particular pixel location.

The homography $\mathcal{H}_k$ in equation (19) in terms of the camera parameters is given by

$$\mathcal{H}_k = K_v \left(R_k + \frac{1}{d_0} T_k [0 \;\; 0 \;\; 1] \right) K_v^{-1} \tag{22}$$

where T_k=$[T_{X_k} \;\; T_{Y_k} \;\; T_{Z_k}]^T$ is the translation vector, and d_0 is the scene depth which is an unknown constant. The rotation matrix R_k is parameterized [30] in terms of θ_X, θ_Y and θ_Z, which are the angles of rotation about the three axes. The camera intrinsic matrix K_v is assumed to be of the form $K_v = \text{diag}(v, v, 1)$, where v is the focal length. Six degrees of freedom arise from T_k and R_k (three each). However, it has been shown in [30] that the 6D camera pose space can be approximated by 3D rotations without considering translations when the focal length is large. An alternate approach [32, 34] is to model out-of-plane rotations by in-plane-translations under the same assumption of a sufficiently long focal length. This is the approach that we also take to reduce the dimensionality of the problem i.e., the set of transformations $\mathbf{T}$ becomes a 3D space defined by the axes t_X, t_Y and θ_Z corresponding to in-plane translations along the X and Y axes, and in-plane-rotations about the Z axis, respectively. The homography given in the equation (22) then simplifies to

$$\mathcal{H}_k = \begin{bmatrix} \cos\theta_{Z_k} & -\sin\theta_{Z_k} & t_{X_k} \\ \sin\theta_{Z_k} & \cos\theta_{Z_k} & t_{Y_k} \\ 0 & 0 & 1 \end{bmatrix} \tag{23}$$

where the translation parameters are given by $t_{X_k} = \frac{vT_{X_k}}{d_0}$ and $t_{Y_k} = \frac{vT_{Y_k}}{d_0}$.

4.2 Single image deblurring

In order to recover the latent image l, our alternating minimization (AM) algorithm proceeds by updating the estimate of the TSF at one step, and the latent image at the next. We minimize the following energy function over the variables $\mathbf{l}$ and $\boldsymbol{\omega}$,

$$E(\mathbf{l}, \boldsymbol{\omega}) = ||(\sum_{k \in \mathbf{T}} \omega(k) \mathbf{H_k}) \mathbf{l} - \mathbf{b}||_2^2 + \alpha \Phi_1(\mathbf{l}) + \beta \Phi_2(\boldsymbol{\omega}). \tag{24}$$

The energy function consists of three terms. The first measures the fidelity to the data and emanates from our acquisition model (21). The remaining two are regularization terms on the latent image $\mathbf{l}$ and the weights $\boldsymbol{\omega}$, respectively, with positive weighting constants α and β that attract the minimum of E to an admissible set of solutions. The regularization terms will be explained in the following sub-sections.

Our algorithm requires the user to specify a rough guess of the extent of the blur (translation in pixels along X, Y axes and rotation in degrees about Z axis) to build the initial TSF. The 3D camera pose space, whose limits are specified by the user, is uniformly sampled to select the initial set of camera poses. We denote this sampled pose space by $\mathbf{S}$ where $\mathbf{S} \subset \mathbf{T}$. In our experiments, the initial TSF contained 200 poses which is still much smaller than the 1500-2000 poses that the whole space $\mathbf{T}$ would contain even for small to moderate blurs.

Note that our algorithm requires no other user input. In contrast, Hu and Yang [34], whose work comes closest to ours, requires the user to input the blur kernels at various locations in the image and we observed that the final deblurring quality depends greatly on the number, location and correctness of these blur kernels. Furthermore, since we model our blur using in-plane rotations and translations, we do not need to know the focal length of the camera as in the case of [30] whose camera pose space is composed of 3D rotations.

In the TSF estimation step, we compute $\boldsymbol{\omega}$ given the current estimate of the latent image $\mathbf{l}$ based on equation (24).

4.2.1 Image Prediction

Similar to [28] we perform an image prediction step at each iteration before TSF estimation to obtain more accurate results and to facilitate faster convergence. The prediction step consists of bilateral filtering, shock filtering and gradient magnitude thresholding. Details of the implementation can be found in [28]. The predicted image, denoted by $\widehat{\mathbf{l}}$, is sharper than the current estimate of the latent image $\mathbf{l}$ and has fewer artifacts.

4.2.2 TSF estimation on a subspace of T

In the first iteration, we optimize over the initial TSF by minimizing the following energy function

$$E(\boldsymbol{\omega}) = ||\mathbf{A}\boldsymbol{\omega} - \mathbf{b}||_2^2 + \beta\Phi_2(\boldsymbol{\omega}) \tag{25}$$

where $\mathbf{A} = \sum_{k\in\mathbf{S}} \mathbf{H_k}\widehat{\mathbf{l}}$ and $\Phi_2(\boldsymbol{\omega}) = ||\boldsymbol{\omega}||_1$. Similar to [28], we work on gradients instead of image intensities in our implementation of equation (25) since image derivatives have been shown to be effective in reducing ringing effects [26]. This optimization problem can be solved using the nnLeastR function of the Lasso algorithm [41] which

considers the additional $l1 - norm$ constraint and imposes non-negativity on the TSF weights. Only the 'dominant' poses in the initial TSF $\mathbf{S}$ are selected as a result of the sparsity constraint imposed by the $l1-norm$ and the remaining poses which are outliers are removed. We now rebuild the set $\mathbf{S}$ for the second iteration so that its cardinality is the same as the initial TSF. The new poses are picked around the selected dominant poses by sampling using a Gaussian distribution. This pose perturbation step is based on the notion that the camera trajectory forms a connected 1D path in the camera motion space and, therefore, the poses close to the dominant ones are most likely to be inliers. In the next iteration, equation (25) is minimized over this new 'active' set of poses. The variance of the Gaussian distribution is gradually reduced with iterations as the estimated TSF converges to the true TSF. Experiments on synthetic and real data show that our pose perturbation step lends robustness to the algorithm and it does not get stuck in local minima. Note that the number of columns in $\mathbf{A}$ equals the cardinality of the set $\mathbf{S}$ which is much less than the total number of poses in $\mathbf{T}$. This allows us to compute the matrix $\mathbf{A}$ at the highest image resolution without running into memory issues. We used a β value of 0.1 for our experiments.

4.2.3 Iimage estimation

In this step, the latent image $\mathbf{l}$ is estimated by fixing the TSF weights $\boldsymbol{\omega}$. The blurring matrix is constructed using only the poses in the active set since the the weights of the poses of the inactive set are zero, i.e. $\mathbf{B} = \sum_{k \in \mathbf{S}} \omega(k) \mathbf{H_k}$ and the energy function to be minimized takes the form

$$E(\mathbf{l}) = ||\mathbf{Bl} - \mathbf{b}||_2^2 + \alpha \Phi_1(\mathbf{l}) \tag{26}$$

We use the regularization term $\Phi_1(\mathbf{l}) = || \bigtriangledown l||_2^2$ in [28] and a conjugate gradient method to solve this problem.

4.3 Experiments

This section consists of two parts. We first evaluate the performance of our algorithm on synthetic data and also compare our results with various state-of-the-art single image deblurring techniques. Following this, we demonstrate the applicability of the proposed method on real images using the challenging VIRAT [42] aerial dataset.

We begin with a synthetic example. A latent image of size 720×600 pixels is shown in Fig. 14(a). In order to demonstrate our algorithm's ability to handle 6D motion using just a 3D TSF, we choose the following 6D TSF space- in-plane translations in pixels: $[-8 : 1 : 8]$, in-plane rotations: $[-1.5^\circ : 0.5^\circ : 1.5^\circ]$, out-of-plane translations: $[0.95 : 0.05 : 1.05]$ on the image plane, and out-of-plane rotations: $[\frac{-4}{3}^\circ : \frac{1}{3}^\circ : \frac{4}{3}^\circ]$.

Figure 14: A synthetic example. Rows 1 and 2: (a) Latent image, (b) synthetically blurred image, (c), (d), (e) deblurred outputs obtained using the state-of-the-art deconvolution methods in [37], [34], [30], respectively, (f) deblurred output obtained using the proposed method. Rows 3,4 and 5: Three zoomed-in patches from the images (a) to (f) demonstrating our algorithm's ability to produce artifact-free deblurred outputs.

Figure 15: Deblurring results on the VIRAT aerial database using the proposed method. The first column contains the blurred frames while the second shows deblurred outputs obtained by our algorithm.

	Xu et al. [37]	Hu and Yang [34]	Whyte et al. [30]	Proposed method
PSNR (in dB)	20.47	19.50	21.68	**25.25**
SSIM [43]	0.550	0.470	0.597	**0.725**

Table 2: Comparison with state-of-the-art methods for the synthetic example in Fig. 14.

To simulate the motion of the camera, we manually generate 6D camera motion with a connected path in the motion space and initialize the weights. The pose weights in the TSF are defined in such a way that it depicts the path traversed by a camera with non-uniform velocity. The camera motion thus synthesized is applied (using the TSF model) on Fig. 14(a) to produce the blurred image in Fig. 14(b). The results obtained using the deblurring techniques in [37] [34]and [30]are shown in Figs. 14(c),(d) and (e), respectively. To obtain the result in Fig. 14(f), the following 3D search intervals were input to our algorithm: in-plane translations in pixels: $[-12 : 1 : 12]$, in-plane rotations: $[-2.5^\circ : 0.25^\circ : 2.5^\circ]$. Our algorithm usually converges within 5 to 8 iterations under the criteria that the recovered image does not change above a threshold between two successive iterations. Note that our result is sharp, free from artifacts and compares closely to the original while the comparison methods have some residual blur. Quantitative comparisons are also provided in Table 2.

It can be seen that the proposed method has higher PSNR and SSIM values as compared to the state-of-the-art. Our algorithm and Hu and Yang [34] take approximately 25 minutes on a 3.4 GHz processor running Matlab. The code of Whyte et al. [30] takes much longer in comparison. Although the Matlab implementation of Xu et al. [37] (approximately 15 minutes) runs faster, the output is not satisfactory.

Next, we test our method on the publicly available VIRAT aerial dataset. Since the database contains videos, we manually extracted some frames to run our algorithm. The frames, shown in the first column of Fig. 15, are at the resolution of the original video i.e., 720×480 pixels. The results obtained using the proposed technique (shown in the second column) clearly demonstrate our algorithm's ability to produce excellent deblurring results even on real data.

5 Conclusions and Future Work

We began by proposing a scheme, which we believe is the first of its kind, to estimate planar orientaion from a single motion blurred image. We revealed the underlying relationship between the surface normal of a planar scene and the induced space-variant nature of blur due to translational motion. Exploiting correspondences among the extreme points of the PSFs, we constructed a set of linear equations whose solution yields the surface normal. The method was validated on synthetic as well as real images. Since our method can explain planes, it offers a convenient platform for attempting restoration of piecewise planar motion-blurred scenes.

Key Accomplishments

- *We revealed how correspondences of extremities of blur kernels in a given observation can be used for estimating plane normal by leveraging the homography relationship among image coordinates lying on the plane. This is the first ever attempt in the area of computer vision to propose correspondence theory within a single image.*

As future work, we plan to relax the translational constraint and extend our method to the case of general camera motion. Yet another direction to pursue is to automatically segment and estimate surface normals when the scene consists of multiple planes.

Next, we proposed a framework to detect changes in blurred images of very large sizes. Traditional deblurring techniques fail to cope with large image sizes, while feature-based approaches for change detection are rendered invalid in the presence of blur. We developed an optimization problem which would perform registration in the presence of blur and detect occlusions simultaneously. We devised an algorithm to choose good sub-images from the large observations to estimate the camera motion thus alleviating issues related to memory and computational resources.

Key Accomplishments

- *A unified framework was proposed for registration and automatic detection of occlude(s) from a motion blurred image pair using a low-rank, sparse error matrix decomposition based on sparsity prior. The method can work on even very large image sizes (100 MB) consuming approximately the same power as required for an image one-tenth the size.*

As future work, we shall consider accommodating illumination variations into our framework.

We also described a methodology for blind restoration of aerial images degraded by blur due to camera motion. Due to the large distances involved, we showed that the space-variant blurred image of the ground plane can be expressed as a weighted average of geometrically warped instances of the original image. Given a single observation, the latent image and its associated warps were estimated within an alternating minimization framework. Several results were given on synthetic data as well as the challenging VIRAT aerial database for purpose of validation.

Key Accomplishments

- *Based on the notion of a global transformation spread function, we proposed a multi -scale scheme to blindly restore space-variant motion blurred images affected by arbitrarily-shaped blur kernels. Compared to state-of-the-art, our multi-scale implementation with sparsity prior offers computational savings of more than 25%.*

Restoration under arbitrary inclination of the planar scene is an interesting topic for future research. Also, the estimated camera motion itself can be potentially exploited as a valuable cue for stabilization.

References

[1] B.J Super and A.C Bovik, "Planar surface orientation from texture spatial frequencies," *Pattern Recognition*, vol. 28, pp. 729–743, 1995. 2

[2] P. Clark and M. Majid, "Estimating the orientation and recovery of text planes in a single image," in *Proc. British Machine Vision Conference (BMVC)*, 2001. 2

[3] H. Farid and J. Kosecka, "Estimating planar surface orientaion using bispectral analysis," *IEEE Transactions on Image Processing, year = 2007, volume = 16, pages = 2154-2160.* 2

[4] Shivani G. Rao T. Greinera and Sukhendu Das, "Estimation of orientation of a textured planar surface using projective equations and separable analysis with m-channel wavelet decomposition," *Pattern Recognition*, vol. 43, 2010. 2

[5] O. Haines and A. Calway, "Detecting planes and estimating their orientation from a single image," in *Proc. British Machine Vision Conference (BMVC)*, 2012. 2

[6] S. McCloskey and M. Langer, "'planar orientation from blur gradients in a single image," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2009. 2

[7] Xu Li and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. European Conference on Computer Vision (ECCV)*, 2010. 2.2.1, 2.3.1, 2.3.1, 1, 2.3.2

[8] David G Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004. 3

[9] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008. 3

[10] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski, "Orb: an efficient alternative to sift or surf," in *Proc. ICCV*. IEEE, 2011, pp. 2564–2571. 3

[11] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004. 3

[12] Chunlei Huo, Chunhong Pan, Leigang Huo, and Zhixin Zhou, "Multilevel sift matching for large-size vhr image registration," *Geoscience and Remote Sensing Letters, IEEE*, vol. 9, no. 2, pp. 171–175, 2012. 3

[13] AP Carleer, Olivier Debeir, and Eléonore Wolff, "Assessment of very high spatial resolution satellite image segmentations," *Photogrammetric Engineering and Remote Sensing*, vol. 71, no. 11, pp. 1285–1294, 2005. 3

[14] Le Yu, Dengrong Zhang, and Eun-Jung Holden, "A fast and fully automatic registration approach based on point features for multi-source remote-sensing images," *Computers & Geosciences*, vol. 34, no. 7, pp. 838–848, 2008. 3

[15] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman, "Removing camera shake from a single photograph," in *ACM Trans. Graphics*, 2006, vol. 25, pp. 787–794. 3

[16] Li Xu and Jiaya Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. ECCV*, 2010, pp. 157–170. 3

[17] Ankit Gupta, Neel Joshi, C Lawrence Zitnick, Michael Cohen, and Brian Curless, "Single image deblurring using motion density functions," in *Proc. ECCV*, 2010, pp. 171–184. 3

[18] Oliver Whyte, Josef Sivic, Andrew Zisserman, and Jean Ponce, "Non-uniform deblurring for shaken images," *International journal of computer vision*, vol. 98, no. 2, pp. 168–186, 2012. 3

[19] Zhe Hu and Ming-Hsuan Yang, "Fast non-uniform deblurring using constrained camera pose subspace.," in *Proc. BMVC*, 2012, pp. 1–11. 3

[20] Li Xu, Shicheng Zheng, and Jiaya Jia, "Unnatural l0 sparse representation for natural image deblurring," in *Proc. CVPR*, 2013, pp. 1107–1114. 3

[21] Abhijith Punnappurath, A.N. Rajagopalan, and Guna Seetharaman, "Registration and occlusion detection in motion blur," in *Proc. ICIP*, 2013. 3, 3.1.2, 3.3, 3.3

[22] Zhe Hu and Ming-Hsuan Yang, "Good regions to deblur," in *Proc. ECCV 2012*. 2012, pp. 59–72, Springer. 3.2.3, 3.3

[23] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al., "A large-scale benchmark dataset for event recognition in surveillance video," in *Proc. CVPR*. IEEE, 2011, pp. 3153–3160. 3.3

[24] Richard J Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam, "Image change detection algorithms: a systematic survey," *Image Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 294–307, 2005. 3.3

[25] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman, "Removing camera shake from a single photograph," in *ACM Trans. Graph.*, 2006, pp. 787–794. 4

[26] Qi Shan, Jiaya Jia, and Aseem Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 73:1–73:10, Aug. 2008. 4, 4.2.2

[27] Li Xu and Jiaya Jia, "Two-phase kernel estimation for robust motion deblurring.," in *Proc. ECCV*. 2010, vol. 6311, pp. 157–170, Springer. 4

[28] Sunghyun Cho and Seungyong Lee, "Fast motion deblurring," *ACM Trans. Graph. (SIGGRAPH ASIA 2009)*, vol. 28, no. 5, pp. article no. 145, 2009. 4, 4.2.1, 4.2.2, 4.2.3

[29] Anat Levin, Yair Weiss, Fredo Durand, and William T. Freeman, "Understanding blind deconvolution algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 12, pp. 2354–2367, Dec. 2011. 4

[30] Oliver Whyte, Josef Sivic, Andrew Zisserman, and Jean Ponce, "Non-uniform deblurring for shaken images," *International Journal of Computer Vision*, vol. 98, no. 2, pp. 168–186, June 2012. 4, 4.1, 4.1, 4.2, 14, 4.3, 4.3

[31] Michael Hirsch, Suvrit Sra, Bernhard Scholkopf, and Stefan Harmeling, "Efficient filter flow for space-variant multiframe blind deconvolution," *Proc. CVPR*, pp. 607–614, 2010. 4

[32] Ankit Gupta, Neel Joshi, C. Lawrence Zitnick, Michael Cohen, and Brian Curless, "Single image deblurring using motion density functions," 2010, Proc. ECCV, pp. 171–184. 4, 4.1, 4.1

[33] Chandramouli Paramanand and Ambasamudram N. Rajagopalan, "Inferring image transformation and structure from motion-blurred images.," in *Proc. BMVC*, 2010, pp. 1–12. 4

[34] Zhe Hu and Ming-Hsuan Yang, "Fast non-uniform deblurring using constrained camera pose subspace," in *Proc. BMVC*, 2012. 4, 4.1, 4.1, 4.2, 14, 4.3, 4.3

[35] Yu-Wing Tai, Ping Tan, and M.S. Brown, "Richardson-lucy deblurring for scenes under a projective motion path," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1603–1618, 2011. 4

[36] Oliver Whyte, Josef Sivic, Andrew Zisserman, and Jean Ponce, "Non-uniform deblurring for shaken images," in *Proc. CVPR*. 2010, pp. 491–498, IEEE. 4

[37] Li Xu, Shicheng Zheng, and Jiaya Jia, "Unnatural l0 sparse representation for natural image deblurring," in *In Proc. CVPR*, 2013. 4, 14, 4.3, 4.3

[38] Neel Joshi, Sing Bing Kang, C. Lawrence Zitnick, and Richard Szeliski, "Image deblurring using inertial measurement sensors," *ACM Trans. Graph.*, vol. 29, pp. 30:1–30:9, July 2010. 4

[39] Y Tai, N Kong, S Lin, and S. Y Shin, "Coded exposure imaging for projective motion deblurring," in *In Proc. CVPR*, 2010. 4

[40] C Paramanand and A. N Rajagopalan, "Non-uniform motion deblurring for bilayer scenes," in *In Proc. CVPR*, 2013. 4.1, 4.1

[41] J. Liu, S. Ji, and J. Ye, *SLEP: Sparse Learning with Efficient Projections*, Arizona State University, 2009. 4.2.2

[42] Sangmin Oh, Anthony Hoogs, A. G. Amitha Perera, Naresh P. Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, J. K. Aggarwal, Hyungtae Lee, Larry S. Davis, Eran Swears, Xiaoyang Wang, Qiang Ji, Kishore K. Reddy, Mubarak Shah, Carl Vondrick, Hamed Pirsiavash, Deva Ramanan, Jenny Yuen, Antonio Torralba, Bi Song, Anesco Fong, Amit K. Roy Chowdhury, and Mita Desai, "A large-scale benchmark dataset for event recognition in surveillance video.," in *Proc. CVPR*. 2011, pp. 3153–3160, IEEE. 4.3

[43] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. 4.3